SISTEMAS OPERATIVOS
Procesos

SUAYED
Ricardo Mancilla Guzmán
410110724
ricardomancillag@hotmail.com
rickman@comunidad.unam.mx

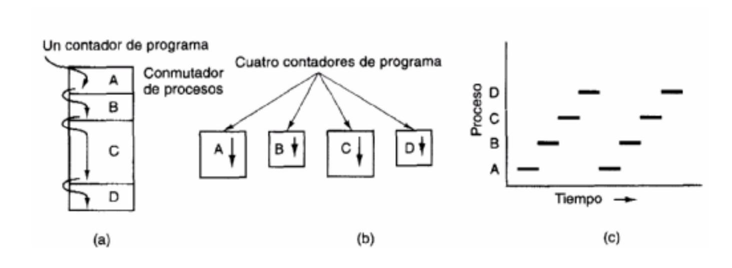
Resumen
El principal programa de un sistema es un sistema operativo, que es el responsable del control de todos los recursos de la computadora y proporciona la base sobre el cual pueden escribirse los programas de aplicación.

Introducción
Es un conjunto de programa que crean la interfaz del hardware con el usuario y que tiene dos funciones primordiales:
Gestionar el hardware, se refiere a hecho de administrar una forma más eficiente los recursos de la máquina.
Facilitar el trabajo al usuario ya que permite una comunicación con los dispositivos de la máquina.

Título Principal

Descripción:
* Qué son los procesos, el modelo de procesos, la creación y terminación de procesos, la jerarquía de procesos y los estados de los procesos.

El modelo de procesos
En este modelo, todo el software ejecutable de la computadora, lo que a menudo incluye al sistema operativo, está organizado en una serie de **procesos secuenciales**, o simplemente **procesos**.
Un proceso no es más que un programa en ejecución, e incluye los valores actuales del contador de programa, los registros y las variables. Conceptualmente, cada uno de estos procesos tiene su propia CPU virtual. Desde luego, en la realidad la verdadera CPU conmuta de un proceso a otro, pero para entender el sistema es mucho más fácil pensar en una colección de procesos que se ejecutan en (seudo) paralelo que tratar de seguir la pista a la forma en que la CPU conmuta de un programa a otro. Esta rápida conmutación se denomina multiprogramación, como vimos en el capítulo anterior.



La creación de procesos se efectúa con FORK, y consta de la serie de pasos.

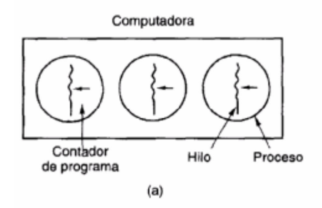
- ✓ Verificar permisos
- ✓ Lee cabecera para obtener los tamaños de segmento total
- ✓ Obtiene los argumentos y el entorno del invocador
- ✓ Asigna nueva memoria y libera la memoria vieja que no se necesita
- ✓ Copia la pila en la nueva imagen de memoria
- ✓ Copia Segmentos de datos en la nueva imagen de datos
- ✓ Verifica y maneja bits stuid y setgid
- ✓ Corrige la entrada de la tabla de procesos

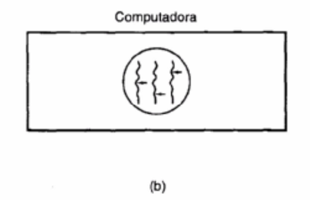
- ✓ Informa el kernel que el proceso es ejecutable

Un proceso termina por completo una vez que han ocurrido dos sucesos: (1) el proceso en sí (o fue cancelado por una señal) y (2) su padre ejecutó una llamada al sistema WAIT para averiguar qué sucedió. Un proceso que ha salido o que ha sido cancelado, pero cuyo padre todavía no ha realizado un WAIT por él, queda en una especie de animación suspendida, a veces EDonúnada estado zombi. El proceso no puede ser planificado y tiene apagado su temporizador alarma (si estaba encendido), pero no se retira de la tabla de procesos.

La parte sigframe proporciona una dirección de retomo para el manejador de señales y datos que SIGRETURN necesita para completar la restauración del estado del proceso cuando el manejador acaba. La dirección de retomo y el apuntador al marco no son utilizados realmente por ninguna parte de MINIX; están ahí para engañar a un depurador por si alguien intentara rastrear la ejecución de un manejador de señales.
La estructura que se colocará en la pila del proceso destinatario de la señal es más o menos grande. El código de las líneas 18225 y 18226 reserva espacio para ella, después de lo cual una llamada a adjust prueba si hay suficiente espacio en la pila del proceso. Si no hay suficiente espacio de pila, se termina el proceso mediante un salto al rótulo determínate empleando el pocas veces usado goto de C (líneas 18228 y 18229).

* Qué son los subprocesos, los hilos y el modelo de los subprocesos.
Los procesos pueden crear subprocesos, que a su vez pueden crear más subprocesos, produciendo un árbol de procesos. De hecho, todos los procesos de usuario del sistema forman parte de un solo árbol con mit en su raíz.
Hilos
En un proceso tradicional del tipo que acabamos de estudiar hay un solo hilo de control y un solo contador de programa en cada proceso. Sin embargo, algunos sistemas operativos modernos manejan múltiples hilos de control dentro de un proceso. Estos hilos de control normalmente se llaman sólo hilos u, ocasionalmente, procesos ligeros.





* Qué es la comunicación entre procesos y la calendarización entre procesos.

COMUNICACIÓN ENTRE PROCESOS
Los procesos con frecuencia necesitan comunicarse con otros procesos. Por ejemplo, en un conducto de shell, la salida del primer proceso debe pasarse al segundo proceso, y así sucesiva mente. Por tanto, es necesaria la comunicación entre procesos, de preferencia en una forma bien estructurada que no utilice interrupciones. En las siguientes secciones examinaremos algunos de los problemas relacionados con esta comunicación entre procesos o IPC.

Cuando se calendariza un proceso, el registro de base se carga con la dirección del principio de su partición, y el registro de límite se carga con la longitud de la partición. A cada dirección de memoria generada se le suma automáticamente el contenido del registro de base antes de enviarse a la memoria. De este modo, si el registro de base es de 100K, una instrucción CALL 100 se convierte efectivamente en una instrucción CALL 100K + 100, sin que la instrucción en sí se modifique.

https://archive.org/details/8891410110724.4.3.1